Nguyen Phan

Dr. Guoning Chen

COSC 6344 Visualization

22 September 2019

**Homework 3 Report**

**Dataset Loading**

I used the C++ code provided within the assignment3.pdf. However, I did not use the build\_edge\_list() function, the edge building process was done within the build\_face\_list() function.

**Data Structures and Storages**

I made various modifications in the provided structs for nodes, lineseg, and quads, instead of holding ints, I made them hold the actual data type. I also added the following structs:

* Intersection: storing data of each intersection (coordinates, parent vertices)
* Intersection\_set: store a vector of intersections for each quad
* contour\_report: store a vector of intersection\_set and the representing scalar value for that iso contour line
* Reports: a vector of contour\_report for storing all iso contour lines

**Displaying Quads**

I implemented display\_quads() function to draw the data by looping through each grid\_pts to draw the quads. Each S data value is input in the previous color scheme functions.

**Algorithm Implementation and Displaying Iso Contour Lines**

The Implementation of the Marching Squares algorithm was first started based on the pseudocode provided within the powerpoint. Then I implemented draw\_line function to draw a line between each two intersections. I first handled only 2 intersection cases, ignoring special cases and 4 intersections. Then I added those special cases into the algorithm.

To display the iso contour lines with the Marching Squares algorithm, I looped through each contour\_report in reports, looped through each intersection set and inputting them into the marching squares algorithm to process each intersection.

**Potential Problems**

At the default perspective (Xrot = Yrot = 0), the lines are displayed with missing segments, I believe this is an issue with a setting within Open GL’s display and perspective, so I had to set the default perspective’s Xrot to 3 for the lines to display properly.